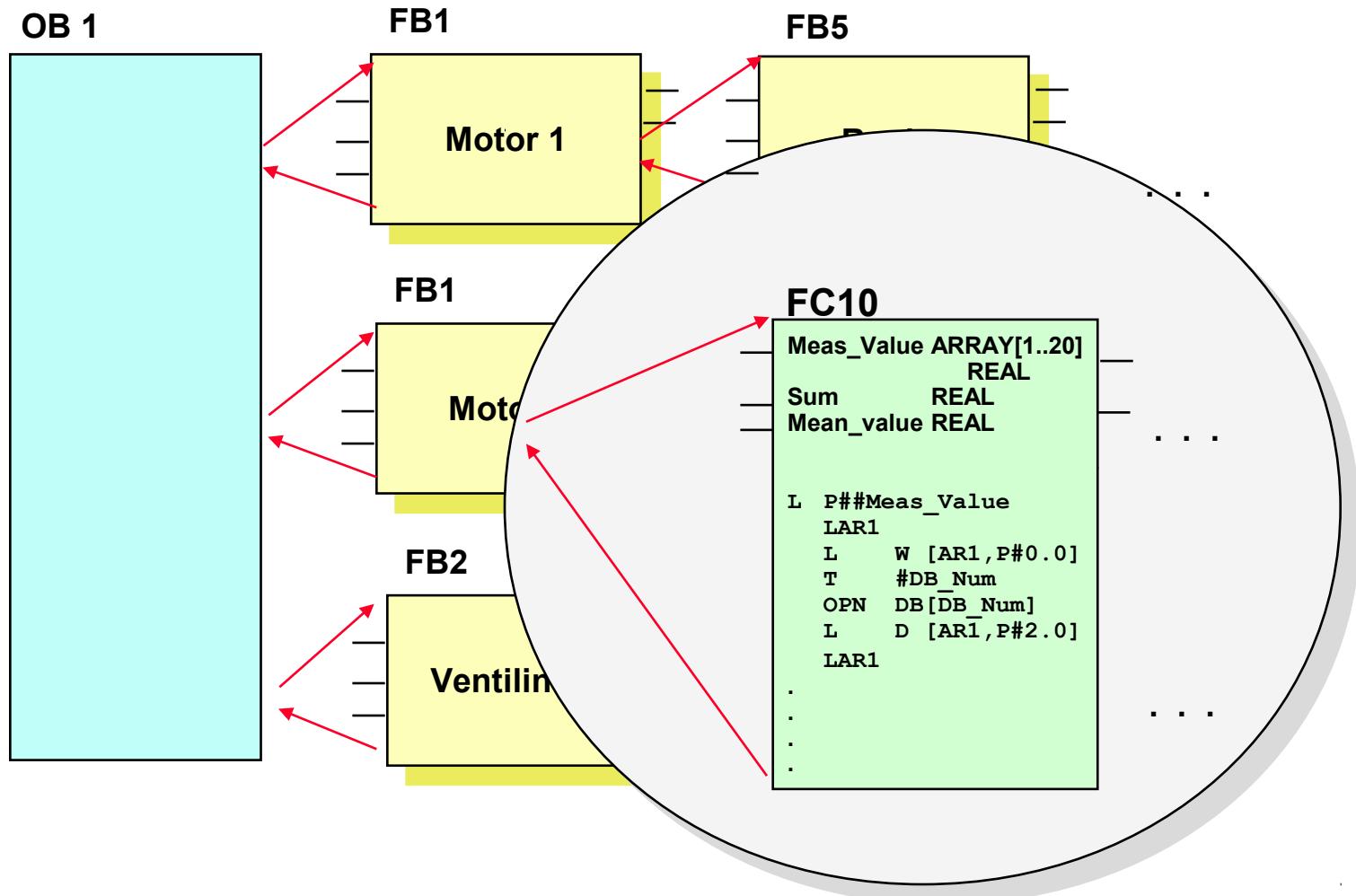
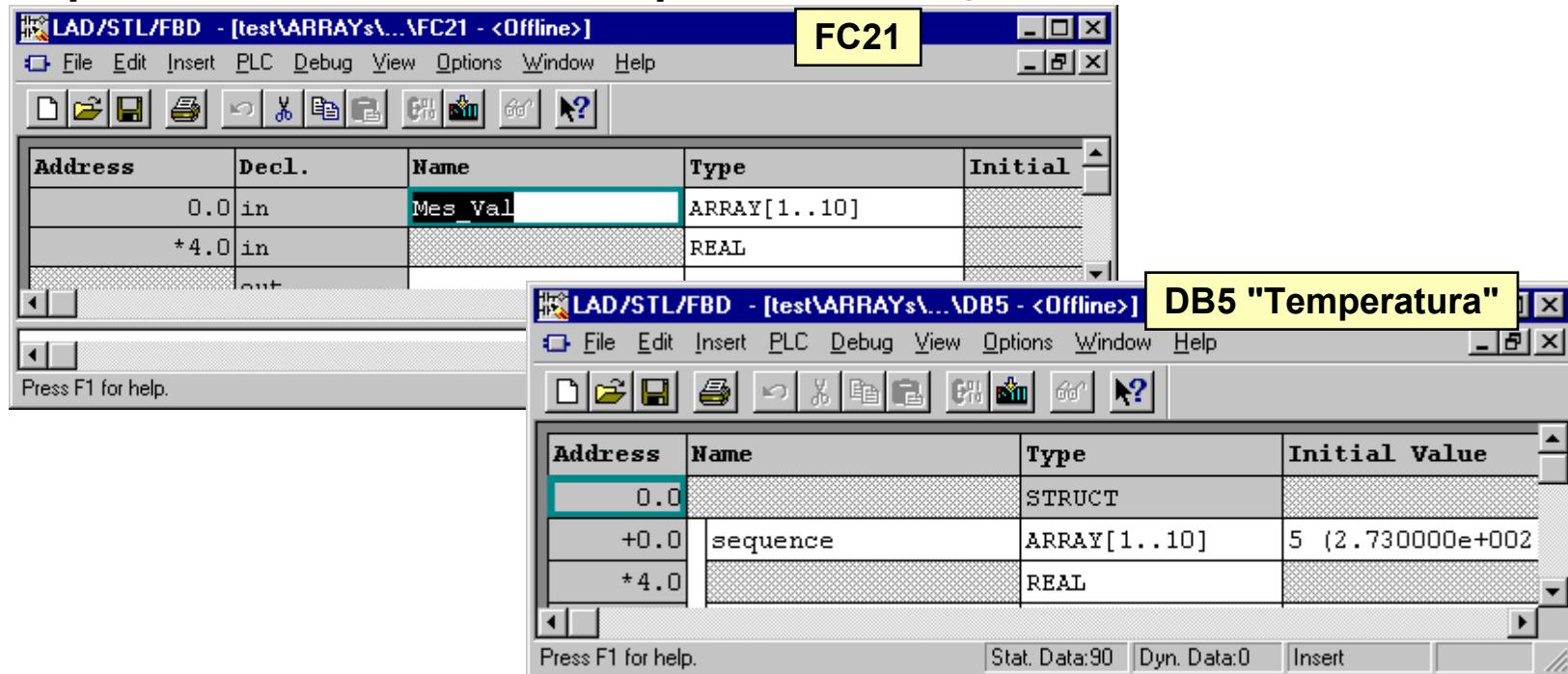


## Apêndice: Acesso Indireto a Parâmetros dos FCs e FBs



# Chamada de Funções com Tipos de Dados Complexos

## Exemplo: Passando um ARRAY para uma Função



A atribuição de parâmetros somente é possível simbolicamente

Network 1: Meas\_Val é declarado como um array na FC21

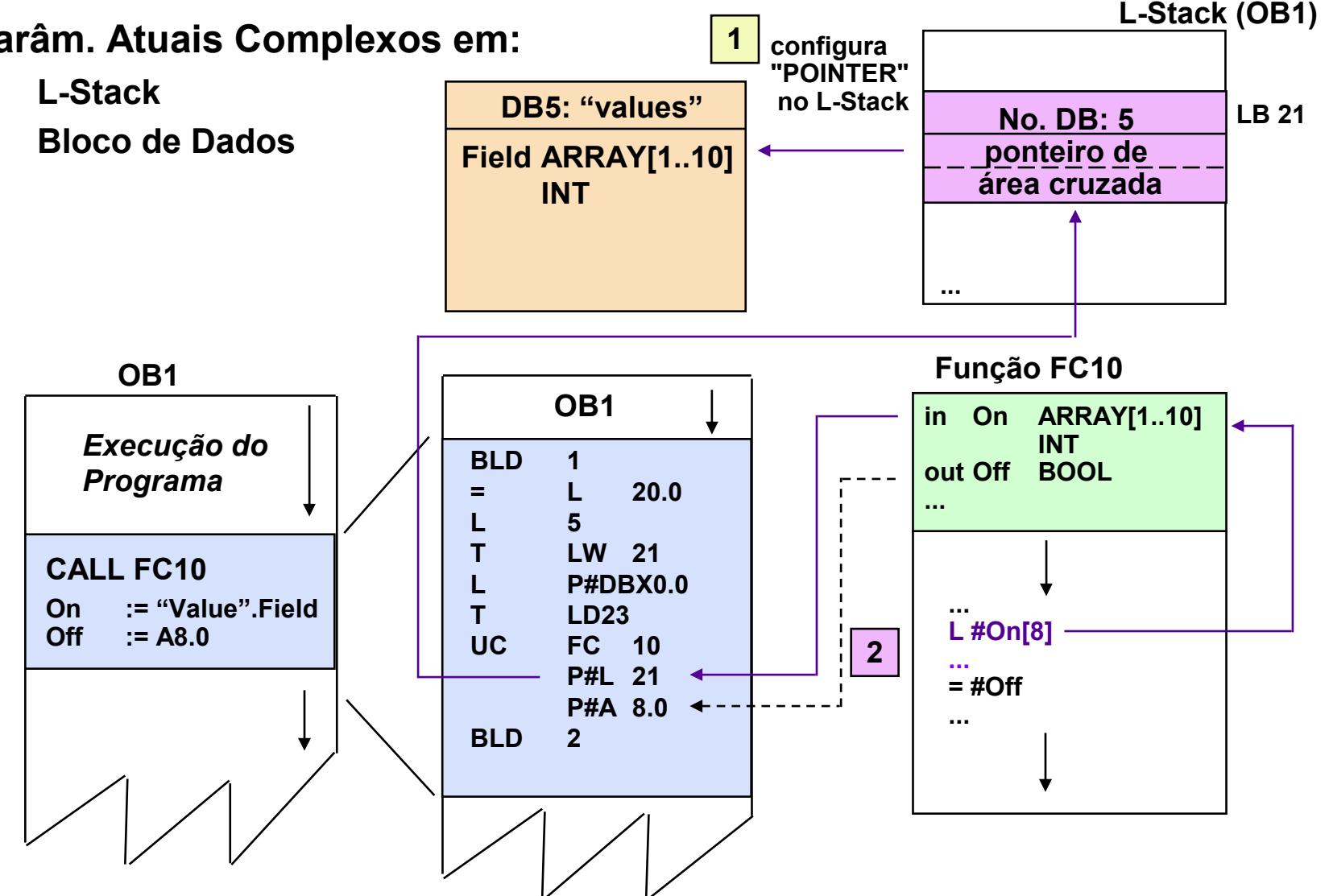
```
CALL FC 21
Meas_Val:="Temperature".sequence
```

# Passando Parâmetros para Tipos de Dados Complexos

Parâm. Atuais Complexos em:

L-Stack

Bloco de Dados



## Acesso Indireto para Tipos de Dados Complexos

Address	Declaration	Name	Type	Start value	Comment
0.0	in	Meas_Val	ARRAY[1..8]		
*4.0			REAL		
32.0	out	Sum	REAL		
36.0	out	Mean_Val	REAL		
	in out				
0.0	temp	DB_Num	WORD		

Network 1: Determining the DB-No. and the start address

```

L      P## Meas_Val           // Carrega endereço do POINTER no ACCU1
LAR1
L      W [AR1, P#0.0]         // e de lá carrega no AR1;
T      #DB_Num               // Determina o número do DB
OPN   DB[DB_Num]             // e carrega na variável temporária;
L      D [AR1, P#2.0]         // Abre DB
LAR1
L      D [AR1, P#2.0]         // Determina área pointer
LAR1
L      D [AR1, P#2.0]         // e carrega no AR1;

```

Network 2: Cálculo da soma (sum)

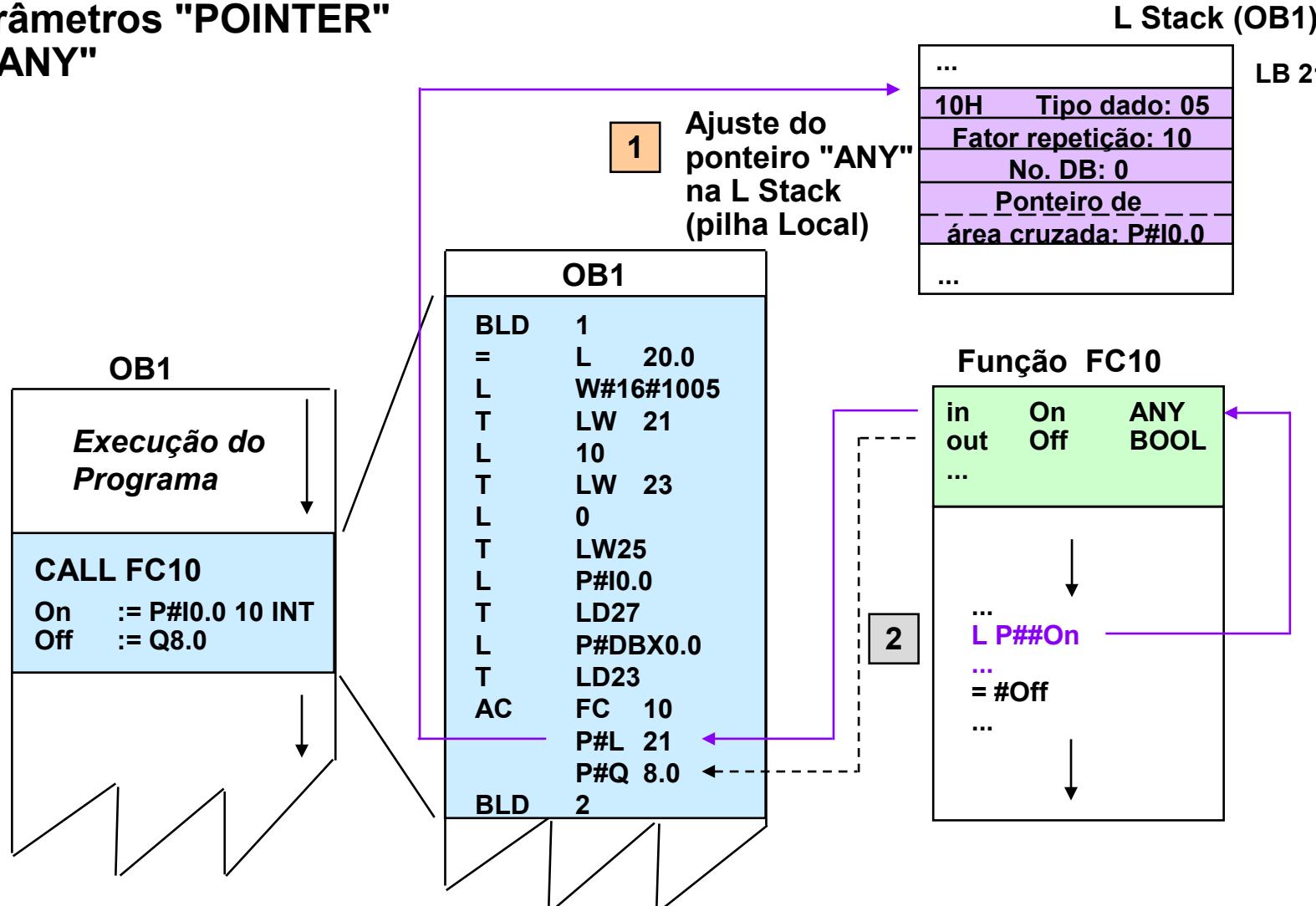
```

L      0.000000e+000          // 0 no ACCU1 (sum =0.0)
L      8                      // Contador para ACCU1; Sum=0 p/ ACCU2
BEGN: TAK
ENT
L      D[AR1, P#0.0]          // Sum p/ ACCU1, contador p/ ACCU2
L      D[AR1, P#0.0]          // Contador p/ ACCU3
+R
+AR1 P#4.0;
TAK
LOOP BEGN;
T      #Sum                  // Decrementa loop contagem, salta se necessário
T      #Sum                  // Transfere sum para #Sum

```

# Passando Parâmetros para Ponteiros

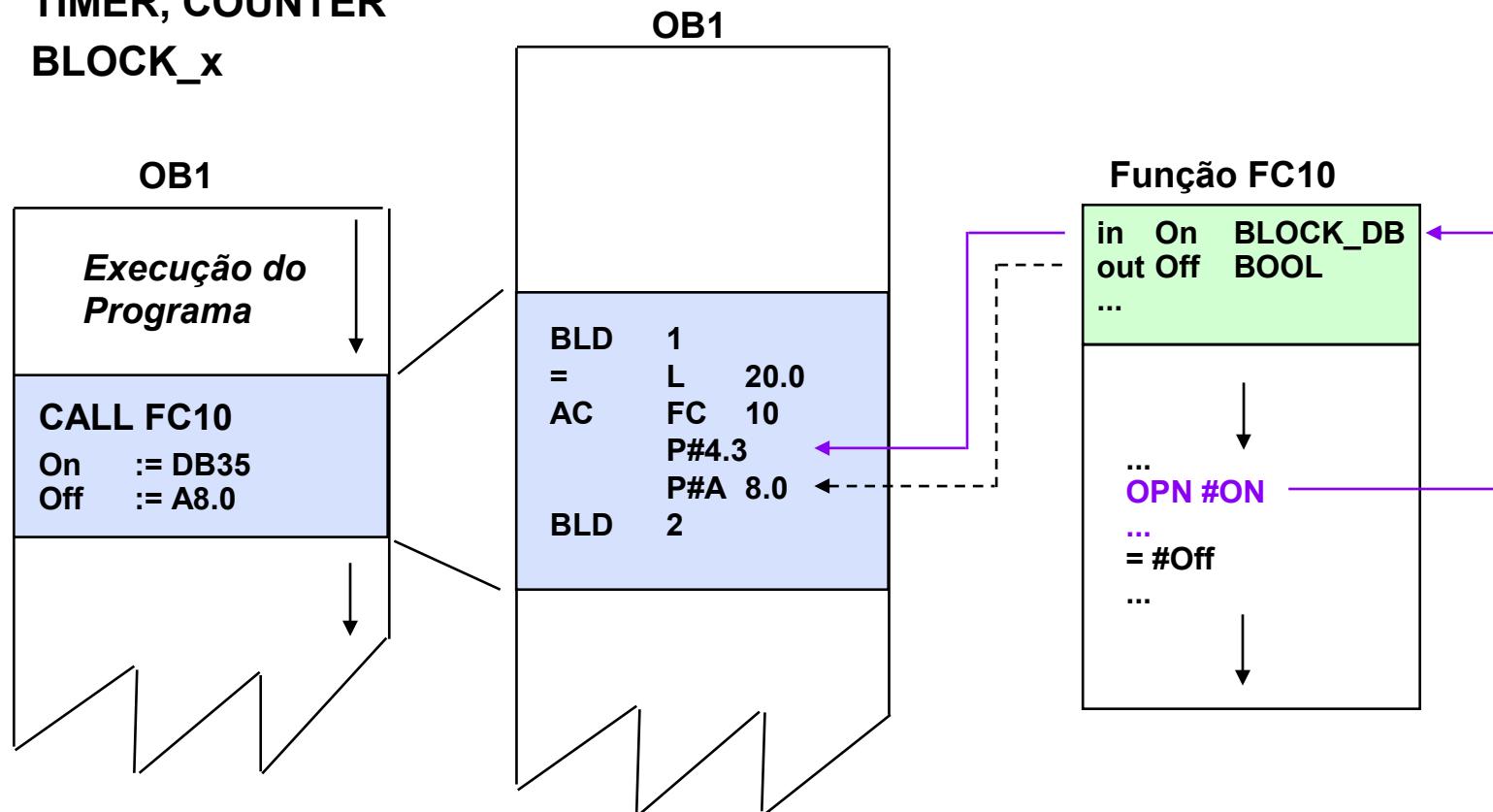
## Parâmetros "POINTER" e "ANY"



# Passando Parâmetros para Tipos de Parâmetros

## Parâmetros de Bloco:

TIMER, COUNTER  
BLOCK\_x

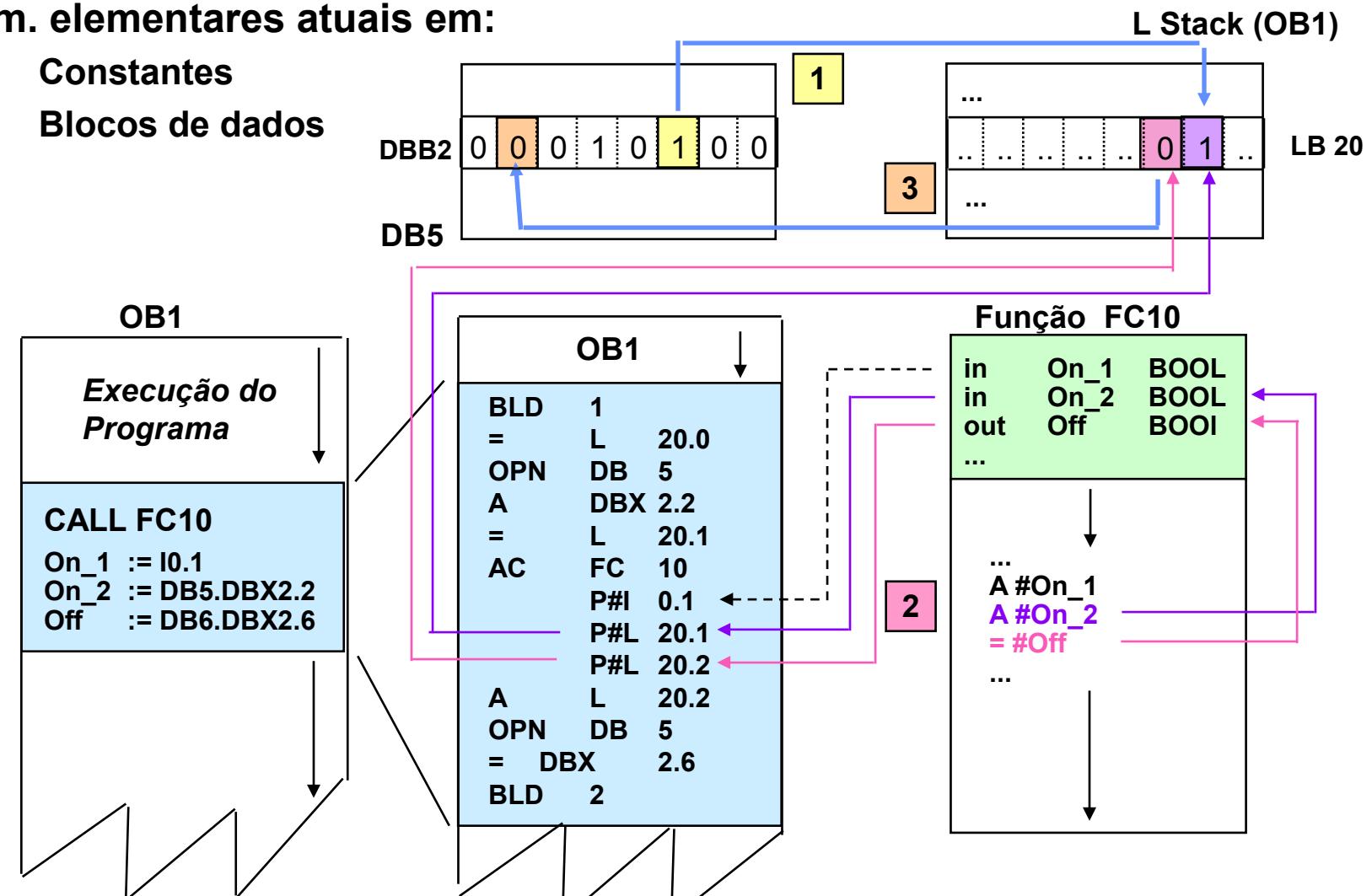


# Construção Especial para Parâmetros Atuais Elementares em DBs e Constantes

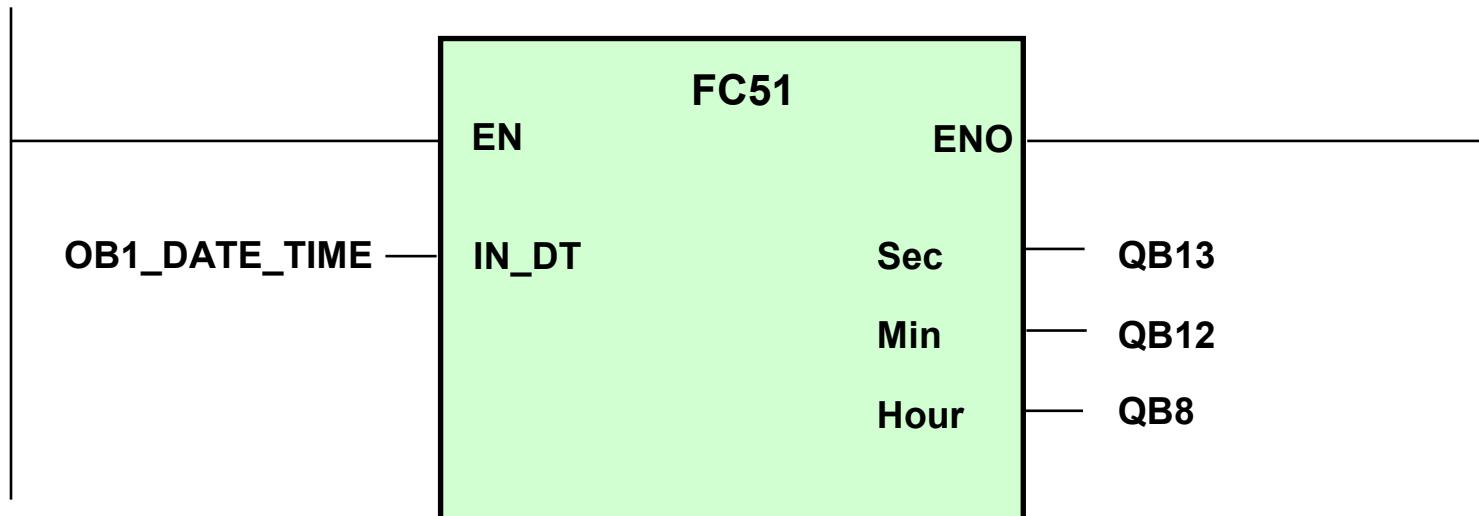
Parâm. elementares atuais em:

Constantes

Blocos de dados



## Exercício A.1: Avaliação do Parâmetro Data e Horário em uma FC



# Chamada de FB Call com Tipos de Dados Complexos

## Exemplo: Passando ARRAYS para um Bloco de Funções

**FB17**

Address	Decl.	Name	Type	Initial Value	Comment
0.0	in	Meas_1	ARRAY[1..10]		
*4.0	in		REAL		
40.0	out	Sum_1	REAL	0.000000e+000	
44.0	out	Sum_2	REAL	0.000000e+000	
48.0	in_out	Meas_2	ARRAY[1..15]		
*4.0	in_out		REAL		
54.0	stat	DB_Num	INT		
58.0	temp				

**DB2 "Temperature"**

Address	Name	Type	Initial
0.0		STRUCT	
+0.0	Cylinder	ARRAY[1..10]	
*4.0		REAL	
+40.0	Shaft	ARRAY[1..15]	
*4.0		REAL	
=100.0		END_STRUCT	

**Atribuindo parâmetros complexos somente é possível simbolicamente**

Network 1:

```

CALL  FB  17,  DB 30
Meas_1   :="Temperature".Cylinder
Sum_1    :=MD20
Sum_2    :=MD30
Meas_2   :="Temperature".Shaft

```

## Acesso Indireto a Parâmetros de Entrada/Saída

Address	Declaration	Name	Type	Start value	Comment
0.0	in	Meas_1	ARRAY[1..10]		
*4.0			REAL		
40.0	out	Sum_1	REAL	0.000000e+000	
44.0	out	Sum_2	REAL	0.000000e+000	
48.0	in out	Meas_2	ARRAY[1..15]		
*4.0	in out		REAL		
54.0	stat	DB_Num	INT	0	

Network 1: Determinando o endereço de início do Meas\_1

```

LAR1 P##Meas_1          // Carrega ponteiro de área cruzada p/parâmetro sem
                        // offset de endereço (multi-instances) no AR1
TAR2
+AR1
                        // Carrega offset de endereço no ACCU1
                        // Soma offset de endereço no AR1;
                        // AR1 agora aponta p/parâmetros no DB instance
                        // DB instance já está aberto

```

Network 2: Acesso para Meas\_1

```

L    0.000000e+000    // 0 no ACCU1 (Soma =0.0)
L    10                 // Contador para ACCU1; Sum=0 p/ ACCU2
BEGN: TAK
ENT
L    D[AR1, P#0.0]     // Campo de componente no ACCU1
+R
+AR1 P#4.0;
TAK
LOOP BEGN;
T    #Sum_1             // Transfere soma para #Sum_1
                        // Decrementa loop de contagem e salta se necessário
                        // Loop de contagem no ACCU1, soma no ACCU2
                        // Contador p/ ACCU3
                        // Soma no ACCU1, contador p/ ACCU2
                        // Incrementa AR1 em 4 bytes
                        // Decrementa loop de contagem e salta se necessário
                        // Transfere soma para #Sum_1

```

## Acesso Indireto a Parâmetros de Entrada/Saída

Address	Declaration	Name	Type	Start value	Comment
0.0	in	Meas_1	ARRAY[1..10]		
*4.0			REAL		
40.0	out	Sum_1	REAL	0.000000e+000	
44.0	out	Sum_2	REAL	0.000000e+000	
48.0	in_out	Meas_2	ARRAY[1..15]		
*4.0	in_out		REAL		
54.0	stat	DB_Num	INT	0	

Network 3: Determinando o endereço inicial do Meas\_2

```

LAR1 P##Messung_2      // Carrega ponteiro de área cruzada p/POINTER sem
TAR2                           // Carrega offset de endereço no ACCU1, soma ao AR1;
+AR1                           // AR1 agora aponta p/POINTER no DB instance
L     W [AR1,P#0.0]        // Carrega número do DB do POINTER no ACCU1
T     #DB_Num               // Transfere número do DB(ou 0) na variável estática
OPN   DB [#DB_Num]         // Abre DB
L     D [AR1,P#2.0]         // Carrega ponteiro de área cruzada p/ parâmetro
LAR1                           // Carrega ponteiro no AR1, AR1 aponta p/parâmetro

```

Network 4: Access to Meas\_2

```

L     0.000000e+000      // 0 p/ ACCU1 (Soma =0.0)
L     15                  // Contador p/ ACCU1; Soma=0 p/ ACCU2
BEGN: TAK                  // Soma no ACCU1, contador no ACCU2
ENT                           // Contador p/ ACCU3
L     D[AR1,P#0.0]         // Campo de componentes no ACCU1
+R                           // Soma no ACCU1, contador p/ ACCU2
...                           // ...

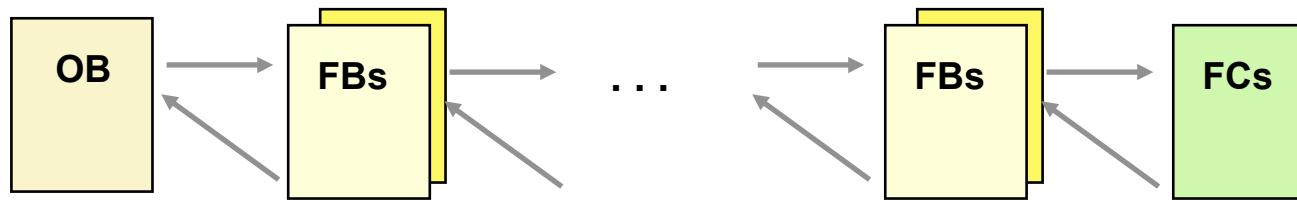
```

# "Passando" Parâmetros

Tamanho do aninhamento:

S7-300: máx. 8

S7-400: máx. 16



A passagem depende do tipo de bloco, dado e parâmetro:

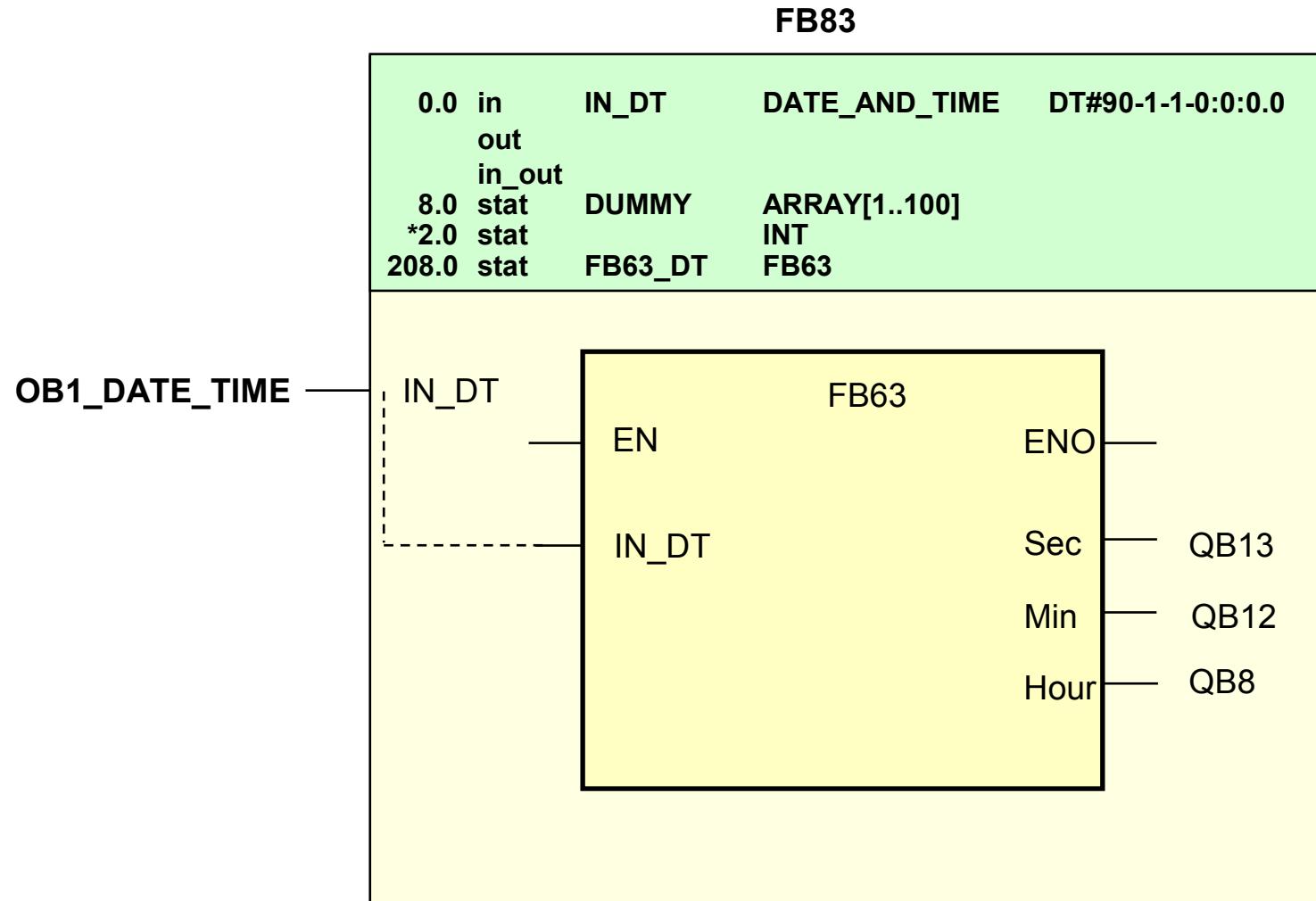
Chamada	FC chama FC	FB chama FC	FC chama FB	FB chama FB
Tipo de dado	P E C	P E C	P E C	P E C
Input -> Input	X - -	X X -	X - X	X X X
Output -> Output	X - -	X X -	X - -	X X -
in/out -> Input	X - -	X - -	X - -	X - -
in/out -> Output	X - -	X - -	X - -	X - -
in/out -> in/out	X - -	X - -	X - -	X - -

E: Tipo de dado elementar

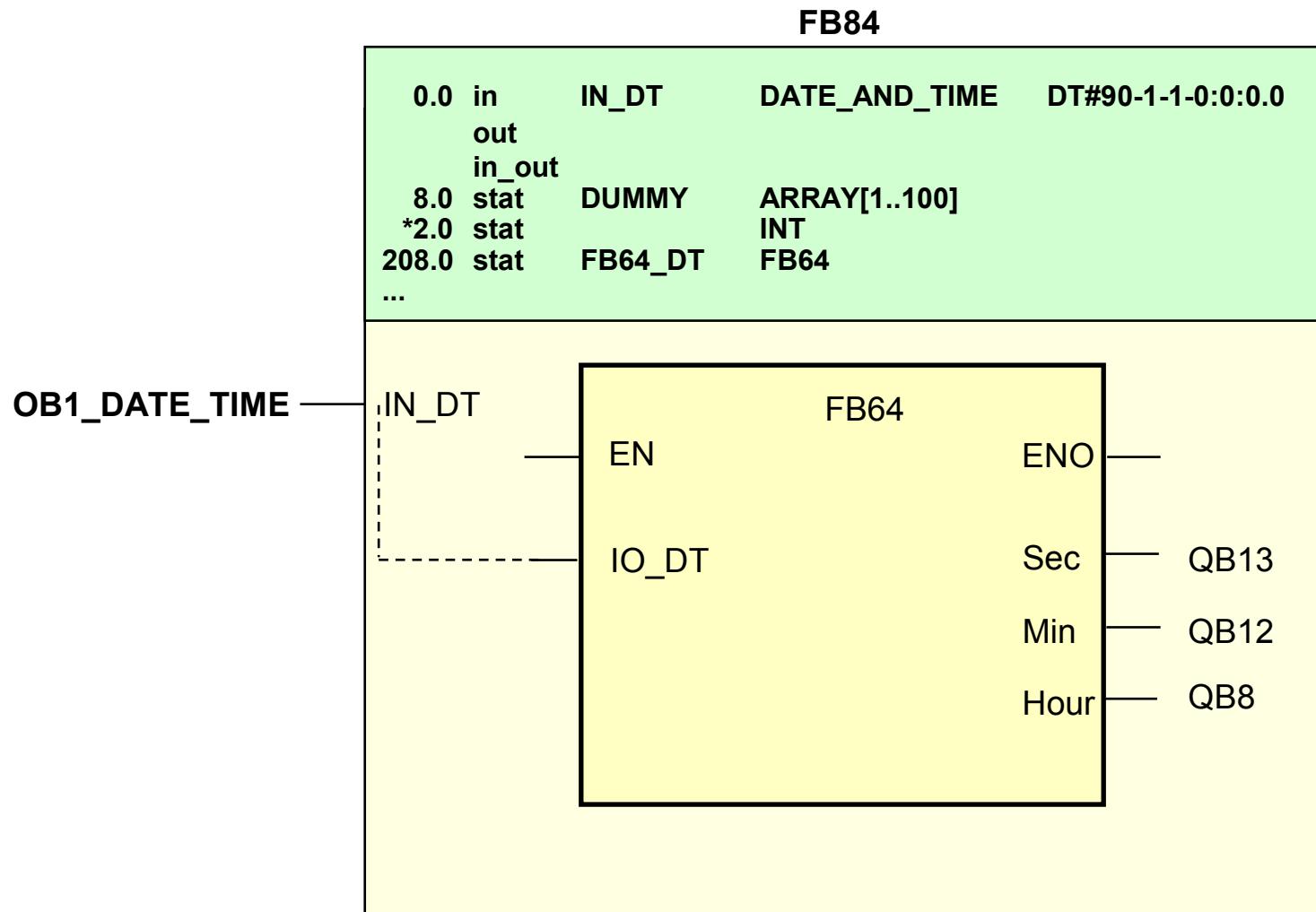
C: Tipo de dado complexo

P: Tipo parâmetro (Timer, Counter, Block\_x)

## Exercício A.2: Avaliação de Parâmetro Data e Horário em um FB



## Exercício A.3: Avaliação de Parâmetros de Entrada/Saída em um FB



## Solução do Exercício A.1: Acesso a Parâmetros DT em uma FC

## Solução do Exercício A.2: Acesso a Parâmetros DT em um FB

## Solução do Exercício A.3: Acesso a Parâmetros Entrada/Saída em um FB (Parte 1)

## Solução do Exercício A.3: Acesso a Parâmetros Entrada/Saída em um FB (Parte 2)